Preceding Rule Induction with Instance Reduction Methods

Osama Othman[1] and Christopher H. Bryant [2]
[1] King Abdullah II School for Information Technology, Jordan University,
Amman, Jordan
Othmano@gi-group.net,
[2] School of Computing, Scienc and Engineering, Newton Building, The University
of Salford, Greater Manchester, M5 4WT, England, UK
C.H.Bryant@salford.ac.uk

Abstract. A new prepruning technique for rule induction is presented which applies instance reduction before rule induction. An empirical evaluation records the predictive accuracy and size of rule-sets generated from 24 datasets from the UCI Machine Learning Repository. Three instance reduction algorithms (Edited Nearest Neighbour, AllKnn and DROP5) are compared. Each one is used to reduce the size of the training set, prior to inducing a set of rules using Clark and Boswell's modification of CN2. A hybrid instance reduction algorithm (comprised of AllKnn and DROP5) is also tested. For most of the datasets, pruning the training set using ENN, AllKnn or the hybrid significantly reduces the number of rules generated by CN2, without adversely affecting the predictive performance. The hybrid achieves the highest average predictive accuracy.

Keywords: Rule Induction, Overfitting, Noise Filtering, Instance Reduction.

1 Introduction

Our work concerns the use of pruning to solve one of the most important problems in the field of machine learning, namely, overfitting which affects the predictive accuracy. We say the produced classifier overfits the data if we can find a different classifier with more error over training examples but smaller error over test data. Overfitting occurs in two situations: when the training set contains noisy instances and when the training set is not a representative sample from the instance space [19]. Both of these situations are common in real world applications.

The aim of our work is to investigate whether overfitting can be reduced by preceding rule induction with instance reduction. We focus on instance reduction methods which have proved capable of reducing the size of training set and resulted in the smallest reduction in predictive accuracy [29], [28]. More specifically, we will apply algorithms that try to remove the border instances, which tend to be noisy instances or hard-to-learn, untypical instances [10].

The paper is organized as follows. Section 2 reviews the typical methods for rule pruning. Section 3 reviews the instance reduction techniques we use in this work. In Section 4, we discuss the results of pre pruning for rule induction using CN2 in terms of predictive accuracy and number of generated rules. Section 5 presents our conclusions.

2 Rule Pruning

A variety of methods has been proposed to prune the produced rule sets, and can be categorized into:

- Prepruning: These algorithms either use heuristics (i.e., stopping criteria.) to relax the constraint of completely satisfying the training instances, or reduce the number of training examples before generating the classifier[9], in the hope that using fewer training examples will produce fewer rules.
- Post Pruning: Initially introduces a rule set that is consistent with training instances, and then the rule set is examined to remove rules and conditions that do not reflect true regularities of the domain. Examples of post pruning algorithms include REP (Reduced Error Pruning algorithm) [2], GROW [5], SSRR [20] and hybrid and incremental post pruning techniques [24].
- Integration Pre Pruning and Post Pruning: Instead of learning the entire rule set and then applying the pruning, this category prunes each rule immediately after it has been learned. Examples of such algorithms are IREP (Incremental Reduced Error Pruning) [11], RIPPER [6], SLIPPER [7] and IREP++ [8].

The aim of our work is to empirically investigate whether pre pruning for rule induction can eliminate some of the produced classification rules while retaining the same level of predictive accuracy.

Pre pruning for rule induction can be achieved in two ways:

1- Condition reductions: pruning each rule independently in the course of learning by using a heuristic to determine when to stop adding conditions to the rule.
2- Rule Pruning: trying to reduce the number of produced rules by either
    a. Removing the most specific produced rules (hopefully that cover the noisy instances from typing or measurement errors).
    b. Reducing the instances used to build the rules.

Previous research on pre pruning focused on simplifying the rules during induction. There is a case study which investigated the effect of a new noisy instance detection method before induction on specific dataset (i.e early diagnosis of rheumatic diseases) [9], and the suggested method is suitable for datasets with just two classes. Grudzinski et al. concentrated on the EkP system [14] as instance reduction method before rule induction, and they illustrated it is possible to extract simpler sets of rules from reduced datasets [13]. However no one has investigated the effect of preceding rule induction with instance reduction methods, in terms of predictive accuracy and number of generated rules.

We will apply some instance pruning methods that have been proven to maintain the predictive accuracy and reduce the size of training set. We will investigate whether applying the rule induction methods to the pruned training set reduces the number of classification rules without adversely affecting the predictive accuracy.

3 Instance Reduction Technique

Instance pruning tries to prune the original training set to get a smaller subset of it. Searching for a subset S of instances to keep instead of the original training set T can proceed in variety of directions, including: incremental, decremental and batch [28].

Incremental methods begin with empty subset S, and add instances (from training set T) to subset S if it fulfills some criteria. Thus if new instances are made available later (after training is completed) they can continue to be added to S according to the same criteria. Incremental methods are sensitive to the order of presentation of the instances. Condensed Nearest Neighbor (CNN) [15] and Selective Nearest Neighbor (SNN) [22] are examples of Incremental methods. On the other hand, decremental methods begin with all the instances in the training set (i.e. T=S), and search for instances to remove; they are often computationally more expensive than incremental methods. Reduced Nearest Neighbor (RNN) [12] and Decremental Reduction Optimization Procedure (DROP1-5) [29] represent examples of decremental methods. Finally, batch methods, as decremental methods, begin with all instances in training set, but before they remove any, they find all of the instances that meet the removal criteria and then they remove them all at once [25]. Batch methods also suffer from increased time complexity compared with incremental methods. In our experiments, we will use decremental and batch methods because, in comparison to incremental methods, they have been shown to give rise to higher predictive accuracies [29].

Instance reduction methods can be categorized as retaining either internal or border instances:

- Border instances: the intuition for retaining border instances is that internal instances do not affect the decision boundaries and thus can be removed with relatively little effect on classification.

- Internal instances: seek to remove border instances, and hopefully removes instances that are noisy.

In our experiments, we focus on three reduction algorithms that performed well in reducing the number of instances [28], and provided good results before applying Neural Network learning [10]. These algorithms eliminate border instances which tend to be noisy instances or hard to learn untypical instances.

3.1 The Edited Nearest Neighbor algorithm

Edited Nearest Neighbor ENN [27] is decremental algorithm which removes an instance if it does not agree with the majority of its k nearest neighbor (with k= 3). This removes noisy instances as well as near border instances and retains all internal instances. Figure 1 shows the pseudo code for ENN algorithm.

3.2 AllKnn

AllKnn [25] is batch algorithm which makes k iteration, at the ith iteration; it flags as bad any instance that is not classified correctly by its i nearest neighbors. After completing all iterations, the algorithm removes all instances flagged as bad. Figure 2 shows the pseudo code for AllKnn algorithm.

```
For each instance (i)
    If (the class of instance (i) <> the majority class of k neighbors)
        Remove the Instance
```

Fig. 1. Pseudo-code for ENN algorithm**.**

```
oldk = k
For each instance (i)
  For k=1 till oldk
    If (the class of instance (i) <> the majority class of k neighbors)
        Flag the Instance for pruning
Remove each flagged instance
```

Fig. 2. Pseudo-code for AllKnn algorithm**.**

```
Let T be the initial set of instances
Measure the distance of each instance in T from its nearest enemy (instance with different class). Sort the instances in T by their distance, in ascending order.
Let S = T.
 For each instance P in S:
    Find $P.N_{1..k+1}$, the k+1 nearest neighbors of P in S.
    Add P to each of its neighbors' lists of associates.
 For each instance P in S:
    Let with= # of associates of P classified correctly with P as a neighbor.
    Let without= # of associates of P classified correctly without P.
    If without >= with
        Remove P from S.
        For each associate A of P
            Remove P from A's list of nearest neighbors.
            Find a new nearest neighbor for A.
            Add A to its new neighbor's list of associates.
 Return S.
```

Fig.3. Pseudo-code for DROP5 algorithm**.**

3.3 DROP5

DROP5 [29] is decremental algorithm which removes the instance "S" if at least as many of its associates (instances have "S" on their nearest neighbor list) are classified correctly without it. It considers removing first the instances that are nearest to their nearest enemy (i.e instance from different class), and proceeding outward. By removing points near the decision boundary first, the decision boundary is smoothed. Figure 3 shows the pseudo code for DROP5 algorithm.

4 Empirical Results for CN2 using the reduced set

Using the noise filtering methods to reduce the border instances before applying the induction method can avoid the overfitting problem. That may improve the predictive accuracy for the induction method. El Hindi and Alakhras (2009) showed that filtering out border instances before training artificial neural network will improve the predictive accuracy and speed up the training process by reducing the training epochs.

The CN2 [4] algorithm induces an ordered list of classification rules from examples, using entropy as its heuristic. Then Clark and Boswell improved CN2 by using a Laplacian error estimate as alternative evaluation function and producing unordered classification rules [3]. Our objective is to apply some Instance reduction methods before applying the modified CN2 algorithm and compare the results with and without applying the reduction

4.1 METHOD

We applied the three methods for instance reduction (AllKnn, ENN and DROP5) that are intended to remove the border and noisy instances before using the CN2. We also apply DROP5 [29] method on instances flagged by AllKnn to be removed and we call this method as AllKnnDROP5 method.

To test if these methods will affect the accuracy of the CN2 algorithm, we conducted experiment on a collection of Machine Learning data sets available from the repository at University of California at Irvine [18]. Predictive accuracy was estimated using 10-fold cross-validation [16]. Instance-removal was performed separately for each fold of the cross-validation.

4.2 RESULTS

Table 1 shows the results obtained using the four prepruning methods with respect to the predictive accuracy. Our experiments show that applying the AllKnnDrop5 algorithm is generally better than applying the other pruning methods with respect to predictive accuracy. Also the results reveal that the predictive accuracy increased on 13 datasets when applying AllKnnDrop5, on 11 datasets when applying AllKnn and on 10 datasets after using the ENN. However the predictive accuracy increased on only 4 datasets after using the DROP5. On average we can see that CN2 after using AllknnDROP5 and AllKnn has better predictive accuracy than CN2 without pruning. This shows that applying a prepruning technique on training set before applying the rule induction can reduce the overfitting problem that can adversely effect on predictive accuracy.

Table 2 shows that all of the instance reduction techniques reduce the number of rules generated by CN2. We can see that DROP5 achieved the largest reduction. Applying AllKnnDrop5 and AllKnn reduces the generated rules by 44% and 48% on average respectively.

From our results we can state that applying instance reduction techniques as a prepruning process for rule induction will reduce the number of generated rules and not adversely affecting the predictive accuracy and may improve it in some cases

Table 1 Empirical results comparing predictive accuracy for using AllKnn ENN, DROP5 and AllKnnDrop5 prepruning**.**

Data Sets	*Without pruning*	*ENN*	*AllKnn*	*DROP5*	*AllKnnDrop5*
Iris	*89.98*	*92.00*	*92.67*	*80.67*	*93.34*
Voting	*95.34*	*95.10*	*95.33*	*85.35*	*95.57*
Vowels	*67.11*	*65.97*	*66.75*	*85.07*	*67.31*
heart (C)	*80.66*	*76.66*	*77.33*	*71.66*	*79.34*
Glass	*64.76*	*58.05*	*61.98*	*51.92*	*66.22*
Liver disorders	*66.77*	*64.11*	*65.64*	*60.3*	*66.52*
Wine	*91.77*	*94.11*	*93.52*	*70*	*95.28*
Pima Indians Diabetes	*74.61*	*73.69*	*76.34*	*73.82*	*76.18*
Promoters	*85.00*	*81.00*	*80.00*	*63*	*80.00*
Hepatitis	*78.65*	*80.00*	*80.00*	*52.67*	*79.34*
Vehicle	*57.85*	*60.10*	*60.71*	*54.99*	*60.10*
pole-and-cart	*61.68*	*63.88*	*66.24*	*62.56*	*63.51*
Blood Transfusion Service Center	*75.68*	*76.61*	*76.35*	*73.11*	*75.96*
Ecoli	*79.10*	*83.31*	*80.91*	*73.34*	*80.90*
Soybean	*86.32*	*82.67*	*83.01*	*63*	*83.32*
ZOO	*92.00*	*87.00*	*90.00*	*81*	*89.00*
Yeast	*48.98*	*55.47*	*56.43*	*51.82*	*56.56*
Led Creator	*72.30*	*72.30*	*71.30*	*68.9*	*71.90*
vertebral_column	*80.96*	*83.21*	*81.28*	*81.28*	*82.24*
Ionosphere	*89.43*	*85.71*	*86.56*	*53.71*	*85.71*
Wave	*69.70*	*70.38*	*70.74*	*67.96*	*71.38*
Balance Scale	*75.30*	*74.70*	*74.34*	*67.1*	*74.34*
Letter recognition	*70.52*	*69.50*	*67.91*	*58.87*	*69.69*
Average	***76.28***	***75.89***	***76.32***	***67.48***	***76.68***

Table 2 Empirical results comparing generated rules for using AllKnn ENN, DROP5 and AllKnnDrop5 prepruning**.**

		ENN		AllKnn		DROP5		AllKnnDROP5	
Data Sets	R_{CN2}	R_{ENN}	R_{ENN} / R_{CN2}	R_{AllKnn}	R_{AllKnn} / R_{CN2}	R_{DROP5}	R_{DROP5}/ R_{CN2}	$R_{AllKnnDROP5}$	$R_{AllKnnDROP5}$/ R_{CN2}
Iris	6.30	3.9	0.62	3.6	0.57	3	0.48	3.6	0.57
Voting	17.3	6.2	0.36	5.7	0.33	3	0.17	6.1	0.35
Vowels	46.2	42.2	0.91	41.5	0.9	31.7	0.69	44.3	0.96
heart (C)	21.3	11.2	0.53	9.4	0.44	7	0.33	10.6	0.5
Glass	22.0	12.8	0.58	12.1	0.55	9.2	0.42	10.3	0.47
Liver disorders	31.3	17.6	0.56	15.2	0.49	12.6	0.4	18.1	0.58
Wine	8.60	7.4	0.86	6.9	0.8	3	0.35	6.9	0.8
Pima Indians Diabetes	44.4	20.8	0.47	18.1	0.41	15.6	0.35	21.3	0.48
Promoters	12.4	10.4	0.84	9.6	0.77	2.7	0.22	9.7	0.78
Hepatitis	17.8	1.80	0.1	4.2	0.24	1.7	0.1	4.7	0.26
Vehicle	48.4	29.3	0.61	25.9	0.54	27.2	0.56	29.3	0.61
pole-and-cart	109.8	56.9	0.52	46.7	0.43	51.7	0.47	50.8	0.46
Blood Transfusion Service Center	61.2	13.0	0.21	11.9	0.19	13.2	0.22	16.5	0.27
Ecoli	24.7	12.7	0.51	10.5	0.43	7.7	0.31	12.3	0.5
Soybean	32.7	15.9	0.49	24.8	0.76	21.3	0.65	27.2	0.83
ZOO	8.70	6.1	0.7	6.3	0.72	6.2	0.71	6.3	0.72
Yeast	121.2	40.7	0.34	37.0	0.31	40.5	0.33	47.3	0.39
Led Creator	79.9	21.8	0.27	19.9	0.25	23.4	0.29	24.3	0.3
vertebral_column	16.7	10.4	0.62	9.1	0.54	6.9	0.41	10.1	0.6
Ionosphere	17.6	6.5	0.37	7.2	0.41	4.9	0.28	9.7	0.55
Wave	204.8	118.0	0.58	102.3	0.5	60.3	0.29	111.6	0.54
Balance Scale	150.1	75.4	0.5	63.0	0.42	21.6	0.14	65.2	0.43
Letter recognition	263.8	232.5	0.88	228.0	0.86	173.8	0.66	233.0	0.88
Average	**59.4**	**33.6**	**0.54**	**31.3**	**0.52**	**23.83**	**0.38**	**33.9**	**0.56**

5 Conclusion

In this paper, we have mentioned different instance reduction techniques, and applied them as preprocessing before CN2 algorithm. Our experiments showed that for most datasets, pruning the training set using AllKnn, ENN or AllKnnDrop5 significantly reduces the number of rules generated by CN2 without adversely affecting the predictive performance. Also applying AllKnnDrop5 gave the best result with respect to predictive accuracy on average. Other instance reduction algorithms, such as C-Pruner [30], conduct instance pruning more carefully, so as to avoid deleting important instances. For future work, we recommend comparing them with ENN, AllKnn and DROP5, to investigate whether the technique of preceding rule induction with instance reduction can be further improved.

Only one rule-induction algorithm was used in our experiments. To investigate how generic the technique is, instance reduction should be applied as a pre-processing step before using other rule induction algorithms and the effect on number of generated rules and prediction accuracy observed.

El Hindi and Alakhras (2009) investigated using instance reduction on Neural Network and they reported good results. We recommend testing instance reduction with other types of classifiers like decision trees.

Acknowledgments

We would like to express our sincere gratitude to Dr. Khalil el Hindi for the useful comments, remarks and the suggested ideas which led us to investigate instance pruning as pre pruning technique for rule induction.

References

1- Aha, D. W., Kibler, D. & Albert, M. K. Instance – based learning algorithm, *Machine Learning*, 6:37-66, 1991.

2- Brunk, C. & Pazzini, M. An investigation of noise-tolerant relational concept learning algorithms. *In Proceedings of the 8th international workshop on Machine Learning*, (1991), pp. 389 – 393. Evanston, Illinois.

3- Clark, P. & Boswell, R. Rule induction with CN2: some recent improvements. In: Kodratoffy (Ed.). Lecture Notes in Computer Science (LNCS). *Proceedings of the sixth European working session on learning*, (1991), vol. 482, pp. 151-163 Springer- Verlag, Portugal.

4- Clark, P. & Niblett, T. The CN2 induction algorithm, *Machine Learning*, 3: 261-283, 1989.

5- Cohen, W. Efficient pruning methods for separate-and-conquer rule learning systems. In: Ruzena Bajcsy (Ed.). *Proceedings of the 13th international joint conference on Artificial Intelligence*, (1993), pp. 988-994. Morgan Kaufmann. Chambery, France.

6- Cohen, W. Fast effective rule induction, 3. Armand Prieditis, Stuart J. RussellIn (Eds.). *Machine Learning: Proceedings of the 12th international conference* (1995), pp. 115-123. Lake Tahoe, California: Morgan Kaufmann.

7- Cohen, W. & Singer Y. A simple, fast and effective rule learner. Jim Hendler, Devika Subramanian (Eds.) *Proceedings of the Sixteenth National Conference on Artificial Intelligence* (1999), pp.335–342. AAAI/MIT Press .Menlo Park, CA, USA.

8- Dain O., Cunningham R. & Boyer S. IREP++ a faster rule learning algorithm. In: Michael w, Umeshwar Dayal, Chandrika Kamath and Davis B. (Eds.). *Proceeding fourth SIAM int. conf. data mining* (2004), pp. 138-146. Lake Buena Vista, FL, USA.

9- Dragan Gamberger, Nada Lavrac, Saso Dzeroski. Noise Elimination in inductive concept learning: A case study in medical diagnosis. Lecture Notes in Computer Science (LNCS). 7th International Workshop, ALT '96 Sydney (1996), vol. 1160, pp. 199-212: Springer-Verlag, Berlin.

10- El Hindi, K. & Alakhras, M. Eliminating border instance to avoid overfitting. In: Antonio Palma dos Reis (Ed.). Proceeding of Intelligent Systems and Agents 2009. pp 93 – 99. IADIS press Algarve, Portugal, 2009.

11- Fürnkranz J. & Widmer G. Incremental reduced error pruning. In Cohen W, & Hirsh H. (Eds.). Proceedings of the 11th International Conference on Machine learning (1994 ML-94), pp.70–77. Morgan Kaufmann. New Brunswick, NJ.

12- Gates, G. W. The reduced nearest neighbor rule. Institute of Electrical and Electronics Engineers Transactions on Information theory IEEE, vol. 18(3): pp. 431-433, 1972.

13- Grudzinski K., Grochowski M., Duch W., "Pruning Classification Rules with Reference Vector Selection Methods", Artificial Intelligence and Soft Computing, Springer LNCS Vol. 6113, 2010, pp. 347-354.

14- Grudzinski K. EkP: A fast minimization – based prototype selection algorithm. In Intelligent Information System XVI, pp. 45-53. Academic Publishing House EXIT. Warsaw,2008.

15- Hart, P. E. The condensed nearest neighbor rules. Institute of Electrical and Electronics Engineers Transactions on Information theory IEEE, vol. 14(3): pp. 515-516, 1968.

16- Kohavi, R. A study of cross-validation and bootstrap for accuracy estimation and model selection. In: Mellish C. (Ed.). In proceedings of 14th international joint conference on artificial intelligence (1995), pp. 1137-1143. Morgan Kaufmann. San Francisco, CA, USA.

17- Lukasz A. & Krzysztof J.. Highly scalable and robust rule learner: performance evaluation and comparison. IEEE Transactions on Systems, Man, and Cybernetics, Part B 36(1) (2006): pp. 32-53.

18- Murphy, P. M. & Aha, D. W. "UCI repository of Machine Learning Data bases", 1994.available by anonymous ftp to ics.uci.edu in the pub/machine-learning-databases directory.

19- Mitchell, T.M Machine Learning, New York, NY, USA McGraw-Hill, 1997.

20- Othman, O. & El Hindi, K. Rule reduction technique for RISE algorithm. Advances in Modeling, Series B: Signal Processing and Pattern Recognition, 47, pp. 2, (2004).

21- Pham D T, Bigot S & Dimov S. A rule merging technique for handling noise in inductive learning, Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science , 218 (C) (2004). pp. 1255-1268

22- Ritter, G. L., Woodruff, H. B., Lowry S. R., and Isenhour T. L. An Algorithm for a Selective Nearest Neighbor Decision Rule. IEEE Transactions on Information Theory, vol. 21, no. 6: pp. 665-669, 1975.

23- Schapire, R. & Singer Y. Improved boosting algorithms using confidence-rated predictions. In: Peter L. Bartlett, Yishay Mansour (Eds.). Proceeding COLT' 98 Proceedings of the eleventh annual conference on Computational learning theory, (1998). pp. 80 -91. ACM press, New York, NY, USA.

24- Shehzad, K.. Simple Hybrid and Incremental Post-Pruning Techniques for Rule Induction. IEEE Transactions on Knowledge and Data Engineering, Issue 99: 1-6, 2011.

25- Tomek, I. An experiment with the edited nearest-neighbor rule. IEEE Transactions on Systems, Man, and Cybernetics, 6(6): pp. 448-452, 1976.

26- Weiss S. & Indurkhya N. Reduced complexity rule induction. In: John Mylopouslos and Ray Reiter (Eds.). In Proceedings of 12th International joint conference on Artificial Intelligence, pp. 678-684. Morgan Kauffmann. Sydney, Australia, 1991.

27- Wilson, D. L. (1972). Asymptotic properties of nearest neighbor rules Using Edited Data. IEEE Transactions on Systems, Man, and Cybernetics, 2(3): pp. 408-421, 1972.

28- Wilsson, D. R. & Martinez, T. R. Instance Pruning Technique. In: Douglas H. Fisher (Ed.). Machine Learning: Proceedings of the Fourteenth International Conference (ICML'97), pp. 403-411. Morgan Kauffmann. San Francisco, CA, 1997.

29- Wilsson, D. R. & Martinez, T. R. Reduction techniques for instance based learning algorithms. Machine Learning, 38(3): pp. 257-286, 2000.

30- Zhao, K. P., Zhou, S. G., Guan, J. H., & Zhou, A. Y. C-Pruner: An improved instance pruning algorithm. Proceedings of the 2th International Conference on Machine Learning and Cybernetics, vol. 1, pp. 94–99, 2003, Sheraton Hotel, Xi'an, China: Piscataway, NJ : IEEE, 2003